USER'S MANUAL

SPACE COAST MICROSYSTEMS

MODEL SCM-512

EPROM PROGRAMMER

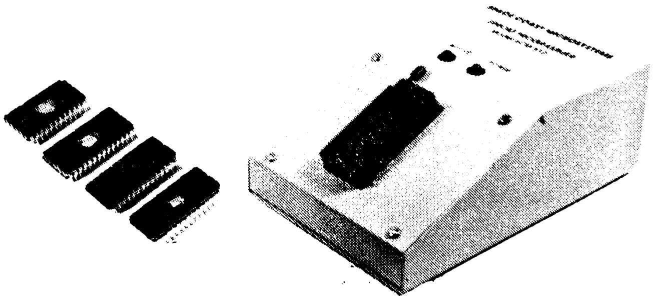

TABLE OF CONTENTS

INTRODUCTION.............................. 1

GETTING STARTED........................... 1

USING CROSSTALK........................... 1

COMMANDS.................................. 2
- Address mode toggle..................... 2
- Blank check............................. 3
- Checksum EPROM.......................... 3
- Delay after C/R transmission............ 3
- Erase 2816A............................. 3
- Menu of device types.................... 4
- Program EPROM from file................. 4
- Quick programming toggle................ 4
- Read EPROM.............................. 5
- Toggle blank check before program....... 5
- Verify EPROM against file............... 5
- Zap (program) individual addresses...... 6
- ?-Print help list....................... 6

FILE FORMATS.............................. 6

USING OFFSETS............................. 8

PROGRAMMING ALGORITHMS.................... 8

COPYING EPROMS............................ 9

COMBINING EPROMS.......................... 9

HANDSHAKING.............................. 10

INTERFACE................................ 10

DEVICE/MANUFACTURER LIST................. 11

ERROR MESSAGES........................... 12

INTRODUCTION

Congradulations. You have just purchased an EPROM programmer with one of the best performance to price ratios on the market today. The SCM-512 supports nearly all 24 and 28 pin devices available. Its fast programming algorithms and flexible command set make it powerful yet very easy to use. The RS232C interface detects any baud rate up to 19200 baud with full software and hardware handshaking. A 48 character FIFO buffer allows data to be sent with as little delay as possible. Programming can be done using standard file formats such as Intel and Motorola, or by programming individual addresses. File offsets can be specified to relocate any file in a 16 bit range.

All the commands of the SCM-512 are executed very quickly. This is due to Z8 microcontroller running at over 7 Mhz. A blank check for a 27256 typically takes 3 seconds. A full help menu for all commands can be displayed at any time by entering a question mark. These and other features make the SCM-512 the most powerful EPROM programmer in its price range.

GETTING STARTED

In order to operate, the SCM-512 must be connected to the standard RS232C DCE interface of a computer or terminal. When power is turned on, the green POWER LED should light and the red ACTIVE LED should be blinking. This indicates that the programmer is waiting for a log-on, not that the ZIF socket is active. If for some reason the red LED is not blinking, the programmer has synced to an invalid baud rate and power must be turned off and on again. To connect to the SCM-512, send a single space at the desired baud rate (300 to 19200) and the copyright text should appear. If this message is not displayed, it may be necessary to connect a null-modem adaptor to correct a reversed RS232 interface (see INTERFACE).

When inserting 24 pin devices into the ZIF socket, make sure they are placed in the bottom section of the socket. If an EPROM is not properly socketed, an error message will be displayed when a command is executed. While the command prompt is displayed, the ZIF socket is completely inactive and any device may be inserted or removed. Take care when executing the P (program) command that the device inserted is selected properly with the M (menu) command.

USING CROSSTALK

When using the IBM terminal program Crosstalk with the SCM-512 be sure to enter the commands GO LOCAL and SP 9600 for local communications. The SE <filename> command is used to transmit a file to the programmer. The CA <filename> is used to capture a file from the programmer to save on disk. If you are using another smart terminal program, consult your users manual for similar commands.

COMMANDS

The SCM-512 enters the command mode upon detection of the baud rate. The current device type can always be seen on the command line. Example: 2764 21V - Command >

Commands to the programmer consist of a single letter optionally followed by one or more parameters. The parameters must be separated from each other by one or more spaces. Any lower case input will be converted to upper case. A carraige return causes the SCM-512 to execute the command line.

All output from the programmer may be halted by a <CNTL-S> and restarted by any character. A <CNTL-C> may be sent at any time to abort a function.

The following table summerizes the commands for the SCM-512. Note that all items enclosed in angle brackets represent optional parameters:

syntax	description
A	Toggle address mode (all/even/odd)
B<str> <end>	Blank check (FF)
C	Checksum EPROM (2 byte sum)
D<dly>	Delay after C/R transmission
E	Erase 2816A
M<A-Q>	Menu of device types
P<str/off> <f>	Program EPROM from file
Q	Quick programming toggle
R<str> <end> <f>	Read EPROM
T	Toggle blank check before program
V<str/off> <f>	Verify EPROM against file
Z<str>	Zap (program) individual addresses
?	Print help list

A) ADDRESS MODE TOGGLE Syntax: A

The A command causes the address modes for the program and verify commands to be changed. It can be used to toggle between three different modes: all, even, and odd. For normal operation, the "all" mode should be used. The SCM-512 defaults to this value on power up.

Some users may want to program one or more seperate EPROM's containing high and low address bytes for a true 16 bit data path. This can be accomplished by programmming one EPROM from a file in the even mode, and another EPROM from the same file in the odd mode. The programming address is generated as follows in these modes:

all: program address = base address
even: program address = base address/2
odd: program address = (base address-1)/2

Any bytes not being even or odd when the mode is selected will not be programmed or verified (an odd address in the even mode would be skipped). Note that if an error occurs, the address displayed is the program address, not the base address.

When in the command mode, a lower case "e" or "o" before the command prompt will indicate that the even or odd address mode is active. Nothing is displayed when the "all" mode is active.

B) BLANK CHECK Syntax: B<str> <end>

A blank check is used to verify that the specified address range contains all erased bytes (FF's). <str> and <end> are the start and end addresses respectively. If <end> is not specified, it will default to the last byte of the currently selected device type. If <str> is not specified, it will default to 0.

Example:	B	Do full blank check.
	B1000 2000	Blank check from 1000H to 2000H.
	B100	Blank check from 100H to end of EPROM.

C) CHECKSUM EPROM Syntax: C

The C command generates a two byte checksum for the socketed EPROM. The sum is generated by taking the data from each address in the EPROM from 0 to the last byte and adding them together to get a 16 bit result. Any carry beyond 16 bits is thrown away. A checksum value is a handy reference to compare an original EPROM to a copy.

D) DELAY AFTER C/R TRANSMISSION Syntax: D<dly>

The D command is used to set the period of time the SCM-512 should wait after it sends a carraige return and line feed. This allows slower terminal programs time to scroll before more data is transmitted. <dly> is the hexadecimal delay in milliseconds up to FFH and defaults to 0. On power up, 20H is assigned for the delay. If you have a fast terminal, setting this to 0 may speed things up.

E) Erase 2816A Syntax: E

The E command is used to erase a 2816A EEPROM. This function is only necessary with a few brands of 2816A devices as most have an automatic byte erase before byte write. The Intel 2816A, for example, requires this chip erase before programming. Other brands such as Xicor, Exel, and Seeq support this as an option, however do not require it. A chip erase should never be attempted on a 2816B or 2864 part!

M) MENU OF DEVICE TYPES

Syntax: M<A-Q>

The M command is used to select the EPROM device type. Just typing M<ENTER> will output a menu list of device types. If the letter corresponding to the device is already known, it can proceed M and the selection will be made without printing the menu.

Example:	MA	Select 2716.
	M	Display selection menu.

The default setting for your unit will be displayed in the command prompt when you first turn it on. Make sure you remember to select the proper device before performing a function.

P) PROGRAM EPROM FROM FILE

Syntax: P<str/off> <f>

The P command is used to program an EPROM from a file. The programming is done in real time as the file is transmitted to the SCM-512. <f> is the letter designating the file format: M for Motorola S, and H for straight hex. If <f> is not specified, the default is Intel hex. <str/off> has two definitions, depending on which file type is used. For Intel hex and Motorola S files, <str/off> is the offset value (see USING OFFSETS). For straight hex files, <str/off> is the start address for programming and defaults to 0.

Once the command line is entered, the SCM-512 will await transmission of the file. A <CNTL-C> will abort back to the command prompt. If an error is encountered, an error message will be displayed (see ERRORS MESSAGES), however the command prompt will not be restored until file transmission is halted.

Example:	P	Program from Intel hex file.
	P+100	Program from Intel hex file, offset forward by 100H.
	P-200 M	Program from Motorola file, subtract 200H from each address.
	PM	Program from Motorola file.
	PH	Program from straight hex file starting at 0.
	P100 H	Program from straight hex file starting at 100H.

Q) QUICK PROGRAMMING TOGGLE

Syntax: Q

The Q command is used to toggle between the regular and quick programming modes. The quick algorithm greatly reduces programming time by using shorter program pulses (see PROGRAMMING ALGORITHMS). Some users may prefer to use the standard algorithms for production EPROM's since very few manufacturers recommend quick programming algorithms at this time.

When the quick programming algorithm is enabled, an asterisk will preceed the command prompt. The SCM-512 defaults to the standard programming mode when turned on.

R) READ EPROM Syntax: R<str> <end> <f>

The R command is used to read an EPROM. <f> is the file format to output: I for Intel hex, M for Motorola S, and H for straight hex. The default file format is a formatted ascii list. <str> and <end> are the start and end addresses respectively. If <end> is not specified, it will default to the last byte of the currently selected device type. If <str> is not specified, it will default to 0. If the list is to be captured to a buffer, open the buffer before pressing <ENTER> on the command line. When the list is completed, the command line will not be restored until a charcter is sent. This keeps the command line text out of the capture buffer.

Example:	R	Read full EPROM in formatted ascii list.
	RI	Read full EPROM in Intel hex format.
	R100 200 M	Read from 100H to 200H in Motorola format.
	R1000 H	Read from 1000H to end of EPROM in straight hex format.

T) TOGGLE BLANK CHECK BEFORE PROGRAM Syntax: T

The T command is used to toggle the blank check before program option on and off. When enabled, any programming attempt on an unerased EPROM will result in an error. This option is enabled when the SCM-512 is powered up. In order to program 28XX series parts which already contain data, this option must be toggled off since erasure is automatically done during programming.

V) VERIFY EPROM AGAINST FILE Syntax: V<str/off> <f>

The V command is used to verify an EPROM against a file. The verification is done in real time as the file is transmitted to the SCM-512. <f> is the letter designating the file format: M for Motorola S, and H for straight hex. If <f> is not specified, the default is Intel hex. <str/off> has two definitions, depending on which file type is used. For Intel hex and Motorola S files, <str/off> is the offset value (see USING OFFSETS). For straight hex files, <str/off> is the start address for verification and defaults to 0.

Once the command line is entered, the SCM-512 will await transmission of the file. A <CNTL-C> will abort back to the command prompt. If an error is encountered, an error message will be displayed (see ERRORS MESSAGES), however the command prompt will not be restored until file transmission is halted.

Example:	V	Verify against Intel hex file.
	V+100	Verify against Intel hex file, offset forward by 100H.
	V-200 M	Verify against Motorola file, subtract 200H from each address.
	VM	Verify against Motorola file.
	VH	Verify against straight hex file starting at 0.
	V100 H	Verify against straight hex starting at 100H.

Z) ZAP (PROGRAM) INDIVIDUAL ADDRESSES Syntax: Z<str>

The Z command is used for programming single addresses without sending a file. The address is displayed and a two character hexadecimal data byte may be entered for programming. <str> is the start address to program and defaults to 0. This command is handy for zeroing out unwanted instructions or characters.

Example: Z Zap addresses starting at 0.
Z60 Zap addresses starting at 60H.

?) PRINT HELP LIST Syntax: ?

The question mark may be entered from the command prompt to display a brief help list of all the commands and their syntax.

FILE FORMATS

Intel hex: Data record

Byte 1 = Colon (3AH)
Byte 2-3 = # data bytes
Byte 4-7 = 16 bit load address
Byte 8-9 = Record type. "00" for data record
Byte 10-x = Data bytes, 2 characters per byte
Byte x+1-x+2 = Checksum, 2 characters
Byte x+3 = Carraige return (0DH)

Intel hex: End record

Byte 1 = Colon (3AH)
Byte 2-3 = "00"
Byte 4-7 = Execution address
Byte 8-9 = Record type. "01" for end record.
Byte 10-11 = Checksum, 2 characters
Byte 12 = Carraige return (0DH)

Intel hex: Extended address record

Byte 1 = Colon (3AH)
Byte 2-3 = "02"
Byte 4-7 = "0000"
Byte 8-9 = "02"
Byte 10-13 = Upper segment base address (USBA)
Byte 14-15 = Checksum, 2 characters
Byte 16 = Carraige return (0DH)

The USBA is the 4 most significant nibbles of a 24 bit address. When specified, it is added to each load address in the file records. The SCM-512 only recognizes the lower 16 bits of the address. The checksum value in an Intel hex file is the two's compliment of the 8 bit sum, without carry, of all the bytes in the record excluding the colon.

Motorola S: Data record, 16 bit address

Byte 1-2 = "S1"
Byte 3-4 = # data bytes + 3
Byte 5-8 = 16 bit load address
Byte 9-x = Data bytes, 2 characters per byte
Byte x+1-x+2 = Checksum, 2 characters
Byte x+3 = Carraige return (0DH)

Motorola S: Data record, 24 bit address

Byte 1-2 = "S2"
Byte 3-4 = # data bytes + 4
Byte 5-9 = 24 bit load address
Byte 10-x = Data bytes, 2 characters per byte
Byte x+1-x+2 = Checksum, 2 characters
Byte x+3 = Carraige return (0DH)

Motorola S: Data record, 32 bit address

Byte 1-2 = "S3"
Byte 3-4 = # data bytes + 5
Byte 5-10 = 32 bit load address
Byte 11-x = Data bytes, 2 characters per byte
Byte x+1-x+2 = Checksum, 2 characters
Byte x+3 = Carraige return (0DH)

Motorola S: End record

Byte 1-2 = "S7", "S8", or "S9"
Byte 3 = Carraige return (0DH)

The SCM-512 only recognizes the lower 16 bits of any load address. The checksum byte for a Motorola S record is the one's compliment of the 8 bit sum, without carry, of all bytes in the record excluding bytes 1 and 2.

Straight hex:

Straight hex files consist of continuous ascii-hex characters optionally seperated by spaces, dashes, or commas. A straight hex file ends with a dollar sign ($) so the SCM-512 knows when to exit back to the command mode.

Example: 00010203040506070809$
00-01,02 03$

USING OFFSETS

It is sometimes necessary to program an EPROM in a different place than an Intel or Motorola hex file specifies. For example, a file assembled to execute at E000H would not directly program into a 2764. The EPROM should be programmed at address 0000H rather then E000H. This is accomplished by using a feature known as the offset.

By proceeding the "P" command with a plus or minus sign, you are telling the SCM-512 to add or subtract an offset to each address in the file. In the above example, the user would want to subtract E000H from each address to relocate the file down to 0000H for programming. The command line would be entered as "P-E000" and then the file can be transmitted to the programmer.

Offset examples:

P+100 = Program from Intel hex file, add 100H to each address.
V-2000 = Verify against Intel hex file, subtract 2000H.
P+1 M = Program from Motorola S file, add 1 to each address.

Note that offsets are not used with straight hex files.

PROGRAMMING ALGORITHMS

The SCM-512 supports four different programming algorithms. To understand them better the following names will be used: STANDARD, QUICK-STANDARD, FAST, QUICK-FAST. The Q command toggles between the regular and quick algorithms.

STANDARD: The standard programming algorithm uses a single fixed pulse to program each address. The following devices support this algorithm: 2716-2732 (50 ms), 2816-28256 (10 ms), 68764 (2 ms), and 2516-2564 (50 ms).

QUICK-STANDARD: The quick-standard programming algorithm applies one sixteenth the normal programming pulse. If the address programs, another pulse is sent for an overprogram. If the address does not program, a full STANDARD pulse is applied. A 2732 typically takes 35 seconds to program under this algorithm.

FAST: The fast algorithm applies 1 millisecond pulses until the adddress programs. When the address verifies, an overprogram pulse of of twice the orignal length is given. A single address can therefore be programmed as fast as 3 milliseconds. The following devices program under the fast algorithm: 2764-27512.

QUICK-FAST: The quick-fast algorithm works exactly like the FAST algorithm except that each pulse is 150 microseconds rather than 1 millisecond. A single address can therefore be programmed as fast as 450 microseconds. A 2764 will typically take 13 seconds to program from an Intel hex file at 19200 baud.

COPYING EPROMS

Making a copy of an EPROM on the SCM-512 requires some form of storage media, such as a disk. The following steps describe how to copy one or more EPROM's using a smart terminal program:

1. Select EPROM type from menu with M command and insert the source EPROM in the socket.
2. Type an R (followed by a start and end address if desired) but do not press <ENTER>.
3. Type the letter "I" to designate an Intel hex file. (the line should read "RI" to read a full EPROM)
4. Open the the storage buffer on your terminal to capture the data to a file.
5. Press <ENTER> and the file will be transmitted.
6. When completed, close the buffer on your terminal then press <ENTER> to restore the command prompt.
7. Insert the destination EPROM in the socket.
8. Type P followed by <ENTER>. The SCM-512 is now in program mode.
9. Transmit the file you captured previously to the programmer.
10. If an error message is displayed, see section on errors.
11. To copy to more than one EPROM, repeat from step 7.

COMBINING EPROMS

It is sometimes necessary to combine two or more EPROM's on a larger capacity EPROM. This is accomplished by reading each of the smaller capacity EPROM's into seperate files and using the file offset feature (see USING OFFSETS) to relocate each file in its proper position in the larger capacity EPROM. The following example shows the steps for combining four 2764 EPROM's into one 27256:

1. Read each 2764 to a different file on disk using a capture buffer (see COPYING EPROMS). The files should be numbered for identification, example: PROM1.HEX, PROM2.HEX, etc.
2. Select and insert the 27256. Use the P command and transmit the first file without an offset. This file will program from 0000H to 2000H. Note that the 27256 has address space from 0000H to 8000H, therefore you have only used one-fourth of the EPROM.
3. The file PROM2.HEX needs to be programmed from 2000H to 4000H. Since it has the addresses 0000H to 2000H in the file, it must be offset forward by 2000H (see USING OFFSETS). The command line should therefore read "P+2000" and the file PROM2.HEX transmitted to the programmer.
4. The file PROM3.HEX needs to be programmed from 4000H to 6000H. As seen from step 3 the command line should read "P+4000" to offset by 4000H.
5. The file PROM4.HEX can then be programmed with the command line "P+6000".
7. Four 2764's have now been combined to fit in one 27256.

HANDSHAKING

The SCM-512 interfaces to a computer or terminal through an RS232 DCE (data communications equipment) interface. Baud rates are supported up to 19200. Although the SCM-512 has an internal 48 character FIFO (first-in first-out) buffer, communication at high speed requires a method of stopping and starting data called handshaking.

CTS/RTS: This method of handshaking uses two lines on the RS232 interface. If CTS is forced low while the SCM-512 is sending data, the programmer will stop transmitting until it returns to a high state. When the internal FIFO of the SCM-512 is nearly full during a file transfer, RTS will go low. Your terminal must stop sending data during this condition (unless XON/XOFF is used) until RTS is brought back high indicating the buffer is close to empty again. DTR is kept high at all times when power is applied.

XON/XOFF: If the RTS line is not monitered, software handshaking can be used. When the input FIFO buffer is nearly full, an XOFF (ascii 19) will be sent from the SCM-512. Transmission must then be halted until an XON (ascii 11) is received. An XOFF (CNTL-S) may also be transmitted at any time to the SCM-512 to halt a long list. Any character may then be sent to start again.

INTERFACE

The back of the SCM-512 has a DB25P connector configured as Data Communications Equipment. If your cable is designed to interface to Data Terminal Equipment you will need to purchase a null modem.

SCM-512:

Pin#	Symbol	Direction	Function
2	(RXD)	<=========	Receive data
3	(TXD)	=========>	Transmit data
4	(CTS)	<=========	Clear to send
5	(RTS)	=========>	Request to send
6	(DSR)	=========>	Data set ready (Always high).
7	(SG)	<========>	Signal ground.
8	(DTR)	=========>	Data terminal ready (Always high).
20	(DCD)	<=========	Data carrier detect (Not used).

DEVICE/MANUFACTURER LIST

In most cases you can use the obvious selection from the menu. The four exceptions are the 27C64, 27C128, 27256, and 27C256 EPROM's. It is recommended that you observe the following list of manufacturers and devices before attempting to program these EPROM's.

SCM-512 firmware version 1.1

Maunfacturer	Part #	Menu selection
AMD	27C64	E
AMD	27256	H
AMD	27C256	H
FUJITSU	27C64	D
FUJITSU	27C128	F
FUJITSU	27256	I
FUJITSU	27C256	I
FUJITSU	27C256A	H
FUJITSU	27C512	J
HITACHI	27C64	D
HITACHI	27256	H
HITACHI	27C256	H
INTEL	27C64	E
INTEL	87C64	E
INTEL	27256	H
INTEL	27C256	H
INTEL	87C256	H
NATIONAL	27C32	B
NATIONAL	27C256	H
NEC	27C64	D
NEC	27256	I
NEC	27C256	I
NEC	27C256A	H
NEC	27C512	J
TI	27C64	E
TI	27C128	G
TI	27C256	H
TI	27C512	J

If you are not sure which type selection to use or are having problems using the given menu selection please call. If we have the data handbooks avalible or you can descibe the programming characteristics of the part we may be able to help.

ERROR MESSAGES

When an error is encountered by the SCM-512, an asterisk is displayed, followed by a number and message indicating the type of error. The number allows automated software to process errors. Some messages will display an address at which the error occurred, followed by the data that was at that location, a dash, and the data that was expected at that location.

Example: *8 Programming error @ 0030 80-88

This error message indicates that a programming attempt was made at address 0030H with the value 88H, but when verified, the value was read as 80H. This might be the result of an unerased . EPROM. The following is a list of all possible errors displayed by the SCM-512. AAAA indicates the error address, PP indicates data present in EPROM, EE indicates expected data.

*1 Illegal command
- The command line entered contained illegal characters or was not in the correct syntax.

*2 Start exceeds end
- A start address in the command line exceeded the end address. This can only occur on the read and verify commands.

*3 Address exceeds EPROM range
- Access was attempted on address not in the range of the currently selected EPROM device. A 2716 for example is limited to 07FFH. See section on USING OFFSETS for details on relocating files.

*4 Checksum error
- During reception of an Intel hex or Motorola S file, an invalid checksum was received for a record. If encountered in the programming mode, the data in the EPROM will most likely be invalid.

*5 Receive buffer overflow
- The internal 48 character FIFO was overrun. This could indicate that your terminal program is not properly handshaking. In this case, the baud rate must be reduced.

*6 Missocketed or bad EPROM
- The EPROM in the ZIF socket is either incorrectly inserted or is damaged.

*7 Not erased @ AAAA PP-EE
- A blank check failed at address AAAA.

*8 Verification error @ AAAA PP-EE
- EPROM contents did not match the received file at AAAA.

*9 Programming error @ AAAA PP-EE
- Programming algorithm failed at address AAAA.